REMARKS

Claims 1, 3-10, 12, 14-17, 21-25, and 31 are pending in the present application. Claim 2 is canceled. Claims 1, 3, 22, 24, and 31 are amended. And claims 32-39 are added. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 1, 4-10, 12, 14-17, 21-25, and 31 under 35 U.S.C. § 103 as being unpatentable over Rowe et al. (US Patent No. 5,737,599), hereinafter referred to as "*Rowe*," in view of Kauffman et al. (US Patent No. 5,857,203), hereinafter referred to as "*Kauffman*." This rejection is respectfully traversed.

With respect to claim 1, the Office Action states:

> As per claim 1, Rowe teaches a method, in an Internet client, of downloading a file, consisting of components, from an Internet server to a client, comprising the steps of:
>
> initiating a download sequence by which each component is transferred using Internet protocol; and reassembling the components into the file using a profile [col.25 lines 55-68].
>
> Rowe does not specifically disclose the components being files. Kauffman teaches a file system in which a file is divided into smaller pieces with themselves are files and a profile (list) that identifies the pieces [see col.6 lines 35-40, col.7 lines 60-65]. It would have been obvious for one of ordinary skill in the art to use the download method of Rowe with the file system of Kauffman because it would have enable efficient downloading of large file over the Internet.
>
> It is apparent that the system as modified would have download each component files and reassemble the component files using the profile.

Office Action, dated 10 April 2001. Claim 1 is amended to include the features of claim 2. The Office Action rejects claims 2 and 3 under 35 U.S.C. § 103 as being unpatentable over *Rowe* in view of *Kauffman*, as applied to claim 1, and further in view of Averbuch et al. (US Patent No. 5,689,825), hereinafter referred to as "*Averbuch*." This rejection is respectfully traversed.

With respect to claims 2 and 3, the Office Action states:

> As per claims 2-3, Rowe does not specifically disclose what happened when download is interrupted. Averbuch teaches a method for transferring file sequences wherein upon interruption of the download sequence, restarting the download at the component affected by the

> interruption [col.6 lines 20-29]. The component transferred prior to the interruption is not re-transferred [col.6 line 29]. Hence, it would have been obvious for one of ordinary skill in the art to combine the teaching of Averbuch with the system as modified because it would have improve the efficiency of the downloading.

Office Action, dated 10 April 2001. Applicant respectfully disagrees. *Rowe* teaches downloading a multi-page electronic document, which could conceivably be a large file. *Kauffman* teaches an object library in which large files are divided into smaller pieces, which are themselves files, and creates a piece map identifying those pieces. *Kauffman*, col. 6, lines 34-39. When a client performs a retrieve function on one of the large files, the server downloads the piece map. The client then decodes the piece map and downloads the individual pieces to reconstitute the large file. *Kauffman*, col. 10, line 58, to col. 11, line 28. Neither *Rowe* nor *Kauffman* teaches or suggests, when a download sequence is interrupted, restarting the download sequence with the piece affected by the interruption, as specifically recited in amended claim 1.

Averbuch teaches a method and apparatus for downloading updated software to portable wireless communication units. The method of *Averbuch* determines download session parameters including a block size and a number of blocks to be transferred. These parameters are downloaded to and stored in the client. *Averbuch* states, "[w]ith knowledge of the size of the original data transfer, the block size, and the remaining number of blocks to be transferred, the data transfer may continue from where it was left off." *Averbuch*, col. 6, lines 24-27. *Averbuch* teaches continuing data transfer of a single file if the transfer is interrupted. However, *Averbuch* provides no teaching or suggestion regarding downloading a plurality of piece files. At best, *Averbuch* teaches what to do if data transfer of a single piece file is interrupted.

Rowe actually teaches away from the presently claimed invention since *Rowe* directs one to downloading only a portion of a single document file rather than downloading, one-by-one, the component files and reassembling the download file using the identifying information in a profile as in the claimed invention. *See In re Hedges*, 228 U.S.P.Q. 685 (Fed. Cir. 1986). Thus, one of ordinary skill in the art would not be motivated to make the changes proposed by the Office Action.

Furthermore, *Averbuch* actually teaches away from the presently claimed invention since *Averbuch* directs one to downloading blocks with a fixed block size rather than dividing the file into pieces, as in the claimed invention. *See In re Hedges*, 228 U.S.P.Q. 685 (Fed. Cir. 1986). Thus, one of ordinary skill in the art would not be motivated to make the changes proposed by the Office Action.

The present invention recognizes the disadvantage of using file transfer protocol (FTP) for download of large files using a device that is likely to be interrupted during download. Neither *Rowe* nor *Kauffman* teaches the problem or its source. Instead, *Rowe* is directed towards multi-page electronic documents and *Kauffman* is directed towards storage of large files in particular file systems. However, *Averbuch* does teach the problem, but offers a very different solution. *Averbuch* actually teaches transferring the file in fixed data blocks and tracking the number of blocks remaining. This solution is very different from the invention recited in claim 1 and would not lead a person of ordinary skill in the art to make the proposed combination. Therefore, one of ordinary skill in the art would not be motivated to combine or modify the references in the manner required to form the solution disclosed in the claimed invention.

The present invention is directed towards dividing a download file into a plurality of component files and, when a download sequence is interrupted, restarting the download sequence with the piece affected by the interruption. Even if *Rowe*, *Kauffman*, and *Averbuch* could be properly combined, the combination would not form the invention recited in claim 1. Instead, a combination of *Rowe*, *Kauffman*, and *Averbuch* would result in a method for downloading a multi-page electronic document, in which the document file is divided into a plurality of piece files that, once downloaded, may be used to reconstitute the document file and, when transfer of one of the piece files is interrupted, the number of blocks remaining is used to restart the transfer of the piece file.

Moreover, the Office Action may not use the claimed invention as an "instruction manual" or "template" to piece together the teachings of the prior art so that the invention is rendered obvious. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Such reliance is an impermissible use of hindsight with the benefit of applicant's disclosure. *Id.* Therefore, absent some teaching, suggestion, or incentive in the prior art, *Rowe*, *Kauffman*, and *Averbuch* cannot be properly combined to form the claimed invention. The presently claimed invention can be reached only through the an impermissible use of hindsight with the benefit of applicant's disclosure a model for the needed changes.

Since claims 3-7 and 32 depend from claim 1, the same distinctions between *Rowe*, *Kauffman*, and *Averbuch* and the claimed invention in claim 1 apply for these claims. Claims 8-10, 12, 14-17, 21-25, 32 and 33-39 are allowable for the reasons stated above with respect to claim 1. Additionally, claims 3-10, 12, 14-17, 21-25, and 31-39 recite other additional combinations of features not suggested by the reference.

Particularly, with respect to claims 6, 10, and 16, the Office Action states:

> As per claims 6, 10, 16, it is well known in the art to use CRC code for identifying file object. A method of producing the unique code would have been a matter of design choice well within the level of one of ordinary skill in the art.

Office Action, dated 10 April 2001. Applicant respectfully disagrees. Cyclical redundancy checking (CRC) codes are typically used for error checking of digitally transmitted data by appending the code onto and sent with a message. The prior art fails to teach or suggest CRC codes in identifying information for each component file of a download file in a profile, which is received before initiating a download of the component files. More specifically, no suggestion to modify *Rowe,* as combined with *Kauffman*, to include such a combination of features has been shown in the Office Action. Instead, this combination of features is dismissed as "a matter of design choice well within the level of one of ordinary skill in the art." Thus, the Office Action fails to establish a *prima facie* case of obviousness.

Further, with respect to claim 7, the Office Action states:

> As per claim 7, Rowe does not specifically disclose verifying the component transferred is part of the file. It would have been obvious for one of ordinary skill in the art to do so because it would have improved the reliability of the system.

Office Action, dated 11 August 1999. However, claim 7 recites "verifying that a component file transferred from the server is part of the download file **using the identifying information**." Therefore, not only does the Office Action casually dismiss the step of verifying that a component file is part of the download file, but the Office

Action fails to address the feature of using the identifying information to verify the component file altogether. Applicant notes that the Office Action neglects to point out any teaching in the prior art that suggests the modification proposed in the rejection. More specifically, the Office Action fails to provide a reference that teaches or suggests "verifying that a component file transferred from the server is part of the download file using the identifying information," as recited in claim 7. Thus, the Office Action fails to establish a *prima facie* case of obviousness.

Therefore, the rejection of claims 1, 3-10, 12, 14-17, 21-25, and 31 under 35 U.S.C. § 103 is overcome.

New claims 32-39 recite that the download sequence is restarted using the profile. *Rowe*, *Kauffman*, and *Averbuch* fail to teach or suggest this feature. *Rowe* fails to teach or suggest dividing a download file into a plurality of component files altogether. *Kauffman* fails to teach or suggest restarting an interrupted transfer. One can only assume that, in the system of *Kauffman*, when a download sequence is interrupted, the client must start from scratch by downloading the piece map and downloading every piece file identified in the piece map. As stated above, *Averbuch* teaches transferring a download file in fixed data blocks, tracking the number of blocks remaining, and continuing data transfer of a single file if the transfer is interrupted. This solution is very different from the invention recited in claims 32-39, wherein the **profile** is used to restart a **download sequence**.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited art and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 10 July 2001

Respectfully submitted,

Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001

APPENDIX

Marked-up Copy of Amended Claims

Please amend claims 1, 3, 22, 24, and 31 as follows:

1. (Three Times Amended) A method, in an Internet client, of downloading a download file, consisting of a set of component files, from an Internet server, comprising:

receiving from the server a profile of the download file that includes identifying information for each component file;

initiating a download sequence by which each component file is transferred, one-by-one, from the server using an Internet protocol;

upon interruption of the download sequence, restarting the download sequence with a component file affected by the interruption; and

when the download sequence is complete, reassembling the component files into the download file using the identifying information in the profile.

3. (Twice Amended) The method as described in Claim [2] 1 wherein any component file transferred prior to the interruption is not re-transferred from the server.

22. (Twice Amended) A client computer connectable to the Internet, comprising:

a processor; and

a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:

receiving from a server a profile of a download file that includes identifying information for a plurality of component files that make up the download file;

initiating a download sequence by which each component file is transferred, one-by-one, from the server using the Internet protocol instructions;

responsive to an interruption of the download sequence, restarting the download sequence with the component file affected by the interruption; and

responsive to receipt of the component files, [for] reassembling the component files into the download file using the profile.

24. (Twice Amended) A server computer connectable to the Internet, comprising:

a processor; and

a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:

breaking a download file into a set of component files;

generating a profile of the download file that includes identifying information for each component file;

transferring the profile to a client;

initiating a download sequence by which the component files are transferred, one-by-one, to the client using the Internet protocol instructions; and

responsive to any interruption of the download sequence, [for] restarting the sequence with the component file affected by the interruption.

31. (Twice Amended) A data processing system, comprising:

a remote control unit; and

a base unit connectable to a monitor for providing Internet access under the control of the remote control unit, the base unit comprising:

a processor; and

a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:

receiving from a server a profile of a download file that includes identifying information for a plurality of component files that make up the download file;

initiating a download sequence by which each component file is transferred, one-by-one, from the server using the Internet protocol instructions;

responsive to an interruption of the download sequence, restarting the download sequence with the component file affected by the interruption; and responsive to receipt of the component files for reassembling the component files into the download file using the profile.

Please add the following new claims:

--32. (New) The method of claim 1, wherein the download sequence is restarted using the profile.

33. (New) The method of claim 8, wherein the download sequence is restarted using the profile.

34. (New) The method of claim 12, wherein the download sequence is restarted using the profile.

35. (New) The computer program product of claim 17, wherein the download sequence is restarted using the profile.

36. (New) The computer program product of claim 21, wherein the download sequence is restarted using the profile.

37. (New) The client computer of claim 22, wherein the download sequence is restarted using the profile.

38. (New) The server computer of claim 24, wherein the download sequence is restarted using the profile.

39. (New) The data processing system of claim 31, wherein the download sequence is restarted using the profile.--